

```

1 ; Ciclo per leggere un carattere da tastiera senza eco e visualizzarlo sulla riga successiva
2 ; L'inserimento dei caratteri termina quando viene inserito il carattere '.' (punto)
3 ; L'inserimento dei caratteri è preceduto dal messaggio "Inserisci caratteri (. =fine)"
4 ;=====
5 ; p302_E1a.asm del libro di testo
6 ;=====
7 .MODEL SMALL
8
9 .DATA
10     car    DB    ?
11     mess   DB    "Inserisci caratteri (. =fine)",0Ah,0Dh,"$" ; Le stringhe da visualizzare
12                                                    ; devono terminare con il simbolo $
13 .STACK                ;direttiva che dichiara il segmento STACK
14
15 .CODE                ;direttiva che dichiara l'inizio del segmento Codice
16
17 Inizio:
18
19     MOV    AX,@DATA ;Inizializzazione del registro del segmento dati
20     MOV    DS,AX
21
22     ;----- programma
23     LEA    DX,mess    ; Visualizza messaggio di inserimento caratteri
24     MOV    AH,09
25     INT    21h
26 ciclo:
27     MOV    AH,07      ; Acquisisce un carattere senza eco
28     INT    21h      ; e il carattere letto viene posto dal sistema in AL
29     MOV    car,AL    ; Salvo il carattere in memoria nella variabile car
30
31     MOV    DL,car     ; Visualizzo il carattere letto
32     MOV    AH,02
33     INT    21h
34
35     CMP    car, "."   ; Confronta se il carattere è '.' (punto)
36     JNE    ciclo     ; Finché il carattere non è '.' ne acquisisco un altro
37                    ; tornando su all'etichetta ciclo:
38
39     MOV    AH,4Ch     ;ritorno al sistema operativo
40     INT    21h
41     END    Inizio

```

```

1 ; Dato un numero A in memoria,
2 ; se A > 10 visualizza "è maggiore di 10"
3 ; se A = 10 visualizza "è uguale a 10"
4 ; se A < 10 visualizza "è minore di 10"
5 ;=====
6 ; p302_E1b.asm del libro di testo
7 ;=====
8 .MODEL SMALL
9
10 .DATA
11     Num      DB 5
12     Magg10   DB "e' maggiore di 10 $" ; le stringhe da visualizzare
13     Min10    DB "e' minore di 10 $" ; devono terminare con il simbolo $
14     Ugua10   DB "e' uguale a 10 $"
15 .STACK      ;direttiva che dichiara il segmento STACK
16
17 .CODE      ;direttiva che dichiara l'inizio del segmento Codice
18
19 Inizio:
20
21     MOV AX,@DATA ;Inizializzazione del registro del segmento dati
22     MOV DS,AX
23     ;----- programma
24     MOV AL,Num ; Porta in AL il contenuto di Num
25     ; ---- Inizio SE
26     CMP AL,10 ; Confronta con 10
27     JA Maggiore ; Se il contenuto del registro AL>10 va all'etichetta Maggiore
28     JB Minore ; Se il contenuto del registro AL<10 va all'etichetta Minore
29     ; --- caso AL=10
30     LEA DX,Ugual10 ; Visualizza messaggio =10
31     MOV AH,09
32     INT 21h
33     JMP fine_se ; salto incondizionato a Fine_se
34 Maggiore:
35     LEA DX,Magg10 ; Visualizza messaggio >10
36     MOV AH,09
37     INT 21h
38     JMP fine_se ; salto incondizionato a Fine_se
39 Minore:
40     LEA DX,Min10 ; Visualizza messaggio <10
41     MOV AH,09
42     INT 21h
43 Fine_se:
44     MOV AH,4Ch ;ritorno al sistema operativo
45     INT 21h
46     END Inizio

```

```

1 ; Ciclo per visualizzare i numeri interi da 0 a 9 in colonna
2 ;   c=0;
3 ;   while(c<10)
4 ;     {cout<<c <<endl;
5 ;       c++;}
6 ;=====
7 ; p302_E1c.asm      del libro di testo
8 ;=====
9 .MODEL SMALL
10
11 .STACK              ;direttiva che dichiara il segmento STACK
12
13 .CODE              ;direttiva che dichiara l'inizio del segmento Codice
14
15 Inizio:
16
17     MOV AX,@DATA   ;Inizializzazione del registro del segmento dati
18     MOV DS,AX
19     ;----- programma
20     MOV CL,0       ; inizializzo CL con il primo valore
21 ciclo:  CMP CL,10   ; Confronto con 10
22         JAE fine_ciclo ; Se CL>=10 si deve uscire dal ciclo (condizione di terminazione)
23
24         MOV DL,CL   ; Porto in DL il valore da visualizzare (una cifra)
25         ADD DL,30h  ; Aggiungo 30h per trasformare la cifra in carattere ASCII
26         MOV AH,02   ; Visualizzo la cifra
27         INT 21h
28
29         MOV DL,0Ah  ; Vado a riga nuova
30         MOV AH,02
31         INT 21h
32         MOV DL,0Dh  ; e mi posiziono a inizio riga
33         MOV AH,02
34         INT 21h
35
36         INC CL      ; valore successivo
37         JMP ciclo   ; Salto incondizionato a inizio ciclo
38
39 fine_ciclo:
40     MOV AH,4Ch     ;ritorno al sistema operativo
41     INT 21h
42     END Inizio

```